

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$)**330.00**

Complete if Known

Application Number **09/833,346**
Filing Date **April 12, 2004**
First Named Inventor **Rodney Carlton Burnett**
Examiner Name **Bagnoc N. To**
Art Unit **2172**
Attorney Docket No. **AUS920010160US1**

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit
Account
Number
Deposit
Account
Name

09-0447

The Director is authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Credit any overpayments

☐ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid
1001 770	2001 385	Utility filing fee	
1002 340	2002 170	Design filing fee	
1003 530	2003 265	Plant filing fee	
1004 770	2004 385	Reissue filing fee	
1005 160	2005 80	Provisional filing fee	
SUBTOTAL (1) (\$)			

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims	Fee from below	Fee Paid
Total Claims	-20** =	X	
Independent Claims	-3** =	X	
Multiple Dependent			

Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description
1202 18	2202 9	Claims in excess of 20
1201 86	2201 43	Independent claims in excess of 3
1203 290	2203 145	Multiple dependent claim, if not paid
1204 86	2204 43	** Reissue independent claims over original patent
1205 18	2205 9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$)

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Small Entity

Fee Code (\$)	Fee Code (\$)	Fee Description	Fee Paid
1051 130	2051 65	Surcharge - late filing fee or oath	
1052 50	2052 25	Surcharge - late provisional filing fee or cover sheet	
1053 130	1053 130	Non-English specification	
1812 2,520	1812 2,520	For filing a request for ex parte reexamination	
1804 920*	1804 920*	Requesting publication of SIR prior to Examiner action	
1805 1,840*	1805 1,840*	Requesting publication of SIR after Examiner action	
1251 110	2251 55	Extension for reply within first month	
1252 420	2252 210	Extension for reply within second month	
1253 950	2253 475	Extension for reply within third month	
1254 1,480	2254 740	Extension for reply within fourth month	
1255 2,010	2255 1,005	Extension for reply within fifth month	
1401 330	2401 165	Notice of Appeal	
1402 330	2402 165	Filing a brief in support of an appeal	330
1403 290	2403 145	Request for oral hearing	
1451 1,510	1451 1,510	Petition to institute a public use proceeding	
1452 110	2452 55	Petition to revive - unavoidable	
1453 1,330	2453 665	Petition to revive - unintentional	
1501 1,330	2501 665	Utility issue fee (or reissue)	
1502 480	2502 240	Design issue fee	
1503 640	2503 320	Plant issue fee	
1460 130	1460 130	Petitions to the Commissioner	
1807 50	1807 50	Processing fee under 37 CFR 1.17(q)	
1806 180	1806 180	Submission of Information Disclosure Stmt	
8021 40	8021 40	Recording each patent assignment per property (times number of properties)	
1809 770	2809 385	Filing a submission after final rejection (37 CFR 1.129(a))	
1810 770	2810 385	For each additional invention to be examined (37 CFR 1.129(b))	
1801 770	2801 385	Request for Continued Examination (RCE)	
1802 900	1802 900	Request for expedited examination of a design application	

Other fee (specify)

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$)**330.00**

SUBMITTED BY

Name (Print/Type) **Darrell Walker**
Signature **Darrell Walker**

Registration No.
(Attorney/Agent)

34945

(Complete if applicable)

Telephone **713-772-1255**

Date **8-2-2004**

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

AF ^{IFW} ~~11~~

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of:
Rodney Carlton Burnett

Application No.: 09/833,346

Filed: April 12, 2001

For: A METHOD FOR CONSTRUCTING
AND CACHING A CHAIN OF FILE
IDENTIFIERS AND ENABLING
INHERITENCE OF RESOURCE
PROPERTIES IN FILE SYSTEMS

Honorable Commissioner
Patents and Trademarks
P. O. Box 1450
Alexandria, VA. 22313-1450

Attorney Docket No.: AUS920010160

Examiner: BAOQUOC N. TO

Group Art Unit: 2172

Express Mail® Mailing Label

Number ER 632199394 US

Date of Deposit: August 2, 2004

I hereby certify that this paper or fee is being deposited with the United States Postal Services "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, P. O. Box 1450, Alexandria, Virginia 22313-1450

Darcell Walker, Reg. No. 34945

APPEAL BRIEF TRANSMITTAL

Dear Sir,

Applicant files this Appeal Brief in support of the Notice of Appeal filed on June 1, 2004 for the above-identified application. It is believed that \$330.00 is due for the filing of this Appeal Brief. Please charge all fees due to Deposit Account Number 09-0447.

Respectfully submitted,

David Mack

Darcell Walker
Reg. No. 34,945
9301 Southwest Freeway
Suite 250
Houston, Texas 77074
713-772-1255

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of:	§	
Rodney Carlton Burnett	§	Attorney Docket No.: AUS920010160
	§	
Application No.: 09/833,346	§	
	§	Examiner: BAOQUOC N. TO
Filed: April 12, 2001	§	
	§	Group Art Unit: 2172
For: <u>A METHOD FOR CONSTRUCTING</u>	§	
<u>AND CACHING A CHAIN OF FILE</u>	§	
<u>IDENTIFIERS AND ENABLING</u>	§	
<u>INHERITENCE OF RESOURCE</u>	§	
<u>PROPERTIES IN FILE SYSTEMS</u>	§	

APPEAL BRIEF

Commissioner of Patents & Trademarks
P. O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir,

I. INTRODUCTION

Applicant appeals to the Board of Appeals and Interferences from the decision of the Examiner mailed on December 29, 2003 finally rejecting applicants' claims under 35 USC §103(a) as being unpatentable over Pinkoski (US Patent No. 5,742,817). Applicant submits this Appeal Brief in triplicate.

II. REAL PARTY IN INTEREST

The real party in interest is the assignee of record

III. RELATED APPEALS OR INTERFERENCES

Applicant is unaware of any appeal or interference proceeding related to the present invention.

IV. STATUS OF THE CLAIMS

Applicant has attached a copy of the pending claims as Appendix 1.

Applicant submitted an amended set to Examiner, on March 29, 2004, in response to the Office Letter received by Applicant mailed on December 29, 2003. Examiner, in an advisory action mailed on April 8, 2004, did not enter the submitted set of amended claims. Therefore, the pending claims are those submitted in the amendment filed on October 9, 2003.

V. STATUS OF AMENDMENTS

Since the Examiner's action dated April 8, 2004 purported to not enter the submitted amendments to the claims, Applicants' understanding is that the claims are pending in this appeal are those claims submitted in Applicant's October 9, 2003, response.

VI. SUMMARY OF THE INVENTION

A. The Problem

In the file system environment, it is difficult to identify the location of attributes that are not in the file system. As a result of the difficulty to locate these external attributes, it is also difficult to access these attributes from within the file system. Existing technology does not adequately and efficiently address this problem.

B. Applicant's Invention

Applicant's present invention, as described in the application, is based on a new technique that involves constructing and caching a chain of FIDs that represent the directory path to a system resource. This invention describes a method for constructing an array of file identifiers "FIDs" also known as a FID chain, which represents all the directory components of a file system pathname to a file system resource. These components represent the individual file system resources in the form of directories, which must be traversed along the file system hierarchy in order to access the file system resource represented by the terminating component in the path. With this method, it is possible to control access to file system resources using externally stored attributes. Furthermore, this invention describes a technique in which the externally stored attribute,

such as an authorization security policy, uses an array of file system identifiers to determine access to a file system resource list in this array.

VII. THE ISSUE

The issue: Is the Applicant's present invention, as claimed, obvious and therefore unpatentable under 35 U.S.C. § 103(a), in view of the descriptions and teachings described in Pinkoski (U.S. Patent No. 5,742,817)?

VIII. GROUPING OF THE CLAIMS

The grouping of the claims is as follows:

Group 1 comprises claim 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 21, 22, 23, 24, 25, 26 and 27.

Group 2 comprises claims 14, 15, 16, 17, 18, 19 and 20.

IX. ARGUMENTS

The Examiner has rejected all of the claims under 35 U.S.C. § 103(a) as being unpatentable over Pinkoski (U.S. Patent No. 5,742,817). Specifically, Examiner concluded that Pinkoski teaches a method for constructing and caching a chain of file identifiers that represent a full path to a file system resource comprising the steps of:

Retrieving a file identifier for the file system resource that corresponds to the processed defined name of the file system resource, this file being the target file identifier in the chain;

Retrieving the file identifier for the next file system resource, said next file resource being the parent for the previous file system resource in the full path

Repeating said retrieving the file identifier for the next file system resource step and said adding the retrieved file identifier to the chain step until a file identifier for each system resource in the full path of the initial file system resource in the chain.

The Examiner further states that Pinkoski does not explicitly teach processing the file system resource's defined name (DN) into a file identifier (FID) and defined name database. The Examiner further states that this step would have been obvious to one of ordinary skill in the art.

Applicant submits that all of the claims are unobvious and patentably distinct from Pinkoski. The Pinkoski patent relates to the internal composition of a file handle (file identifier) and how a file system implementation constructs and decomposes it to efficiently locate an individual object in a file system. In short, the Pinkoski invention is about file handle internals. Referring to column 1, lines 51 through 57 discuss the disadvantages of large file systems that require significant time to perform searches. This significant time requirement has an adverse effect on the overall operation of the server. Pinkoski incorporates an addressing system in which the file handle comprises file system identification, a file identification or node number and a generation number (indicates the focus on the internal composition of the file handle). The implementation of the Pinkoski invention involves the converting of the multi-bit file identification number into an alternative path name that identifies a location for the file without the need for a conversion to a real path name.

Applicant's present invention does not focus on the internal composition of a file identifier, but the use of file identifier to locate non-file system resources. This invention describes how a non-file system implementation utilizes file identifiers (file handles) to represent, what is in effect, a related collection of file system objects (a chain) along a file pathname space leading to the terminal object of the path. As opposed to Pinkoski, Applicant's present invention describes a method that uses file identifiers to construct a file identifier path to a terminal object. The focus of Applicant's present invention is not on the internal composition of the file handle, but on the use of multiple file identifiers to construct a path to a terminal object. Further, the present invention incorporates a technique that caches the constructed multiple file identifiers chain and uses the chain to implement inheritance properties to the terminal object and objects residing above it towards the root (start) of the chain.

Both, the Pinkoski patent and Applicant's present invention describe what file identifiers are and how file systems use them. Pinkoski goes on to claim an invention on the composition of an individual file handle and its use. Applicant's present invention claims as the invention, techniques that use a group of file handles opaquely for collection (chain) of related file system objects, caching that collection and further using it for the purpose of implementing inheritance of properties.

Applicant would like to address the particular rejections in the Examiner's letter dated December 29, 2003. With regard to the step of "Retrieving a file identifier for the file system resource that corresponds to the processed defined name of the file system resource, this file being the target file identifier in the chain", the Examiner cites col.4, lines 60-65. This section of Pinkoski does not discuss or mention the techniques of collecting multiple file identifiers into chains that represent multiple related file system objects according to Applicant's present invention. Rather, Pinkoski on column 4, lines 34 through 65 discusses the use of hexadecimal digits extracted from an individual file handle's contain data to facilitate efficient location of a single file within a file system. There is no mention of constructing and caching any sort of file identifier chains in the section of Pinkoski.

With regard to the step of "Retrieving the file identifier for the next file system resource, said next file resource being the parent for the previous file system resource in the full path", the Examiner cites col. 5, lines 25-30. Again this cited section does not discuss the retrieving of the file identifier for the next file system resource as recited in Applicant's present invention. This section of Pinkoski appears to discuss an internal directory organization for decoding the extracted values from the file handle.

With regard to the step of "Repeating said retrieving the file identifier for the next file system resource step and said adding the retrieved file identifier to the chain step until a file identifier for each system resource in the full path of the initial file system resource in the chain", the Examiner cites col. 5, lines 1-31. Again, the discussion in this section of Pinkoski does not discuss or mention the elements of this step of the claim.

In addition, col. 5, lines 45-49 cited by the Examiner as suggesting the repeating step also fails to discuss or mention the contents of that step of the present invention.

Pinkoski describes addressing schemes for files in server environments. The file handle in a call that accesses a file includes a file system identification, a file identification or inode number and a generation number. Applicant's present invention involves constructing and caching a chain of FIDs (File Identifiers) that represent the directory path to a system resource. Applicant does submit that both inventions have some common objectives of making file retrieval easier. As a result, some portions of each method may appear to be similar. However, the specific implementation of each

method is different. Furthermore, Pinkoski does not even mention FID chains or caching in its disclosure and description.

X. EXAMINER'S REVIEW

In the advisory action issued by the Examiner dated April 8, 2004, Examiner did not enter amendments made to claims after the final rejection. Examiner stated that the inclusion of the step "placing the constructed chain of file identifier for the initial file system resource in a cache" needs further search and consideration. Applicant submits that the feature of the file identifier being placed in a cache was expressly included in claims prior to this amendment. First the preambles of independent claims 1, 14, and 27 describe a method for constructing and caching a chain of file identifiers. Applicant submits that the inclusion of a step expressly stating the insertion of a file identifier chain into a cache should not require the need for additional consideration. Original claim 24 expressly provides for the inserting of a file identifier into the file identifier cache. In addition, claims 3, 10, 11, 15, 16, 19, 21, 26 and 27 originally mentioned the implementation of a cache in Applicant's present invention. Therefore, Applicant submits that the feature of a cache was present during the initial examination and should not require an additional search.

Examiner issued a final rejection in the response dated December 29, 2003. According the section 706.07(a) of the MPEP, under the present practice, second or any subsequent actions on the merits shall be final, except where the examiner introduces new ground of rejection that is neither necessitated by the applicant's amendment of the claims nor based on information submitted in an information disclosure statement filed during the period set forth in 37 CFR. 1.97(c) with the fee set forth in 37 CFR 1.17(p).

Examiner in this case cited a rejection based on new grounds. Examiner issued a 35 U.S.C. 103(a) rejected based on a newly cited reference (U.S. Patent 5,742,817). The examiner did not previously list this reference in any correspondence with the applicant.

In view of section 706.07(a) of the MPEP, applicant submits that the issuance of a final rejection in this office letter dated December 29, 2003 in this case was improper. Applicant requests that Examiner withdraw the final rejection in this case.

XI. CONCLUSION

Under 35 U.S.C. § 103(a), a patent may not be obtained..., if the differences between the subject matter to be patented and the prior art are such that the subject matter as a whole would have been obvious...to a person having ordinary skill in the art to which said matter pertains. Applicant submits that the differences between Pinkoski and the Applicant's present invention were not obvious to one of skill in the art at the time the present invention was made. The present invention has a different focus from the Pinkoski invention with regard to file identifiers and implements the file identifier in a different manner than Pinkoski. In addition, Pinkoski attempts to produce an alternative to the file path name, which is not the case of the present invention. In order for there to be obviousness, there must be a teaching or suggestion in the cited reference that will lead a person of ordinary skill in the art to produce the new invention. In the present case, Pinkoski teaches a technique that focuses on the internal composition of a file identifier. Pinkoski does not teach, suggest or even mention the techniques of Applicant's present invention.

The examiner's rejection of applicants' claims 1 through 27 in this case under 35 U.S.C. § 103(a) are improper. Applicants therefore request the Board to reverse the final rejection by the examiner of claims 1 through 27 and remand the application for further prosecution.

Respectfully submitted,



Darcell Walker

Reg. No. 34,945

9301 Southwest Freeway
Suite 250
Houston, Texas 77074
713-772-1255

APPENDIX I

PENDING CLAIMS ON APPEAL

Claim 1. A method for constructing and caching a chain of file identifiers that represent a full path to a file system resource comprising the steps of:

- processing a file system resource's defined name (DN) into a file identifier (FID) and defined name database;

- retrieving a file identifier for the file system resource that corresponds to the processed defined name of the file system resource, this file identifier being the target file identifier in the chain;

- retrieving the file identifier for the next file system resource, said next file resource being the parent of the previous file system resource in the full path;

- adding the retrieved file identifier to the chain; and

- repeating said retrieving the file identifier for the next file system resource step and said adding the retrieved file identifier to the chain step until a file identifier for each system resource in the full path of the initial file system resource in the chain.

Claim 2. The method as described in claim 1 further comprising after said repeating step the steps of:

- retrieving a file identifier corresponding to the file system resource, which is the target of an access attempt and a chain file identifier representing the full path directory of the target system resource;

- searching for the effective security classification category and defined name for the target resource file identifier;

- updating the security classification system, when said search finds a security classification category for the target resource file identifier;

- determining whether operations for the target file system resource could affect the file system name space; and

- terminating said method when operation does not affect the file system name space.

Claim 3. The method as described in claim 2 further comprising the step of flushing a file identifier chain cache when there is a determination that desired operations on the target file system resource could affect the file system name space.

Claim 4. The method as described in claim 2 further comprising before said file identifier (FID) retrieval step the step of processing the defined name (DN) of a system resource and security classification category into a mapping database which holds a FID to DN mapping.

Claim 5. The method as described in claim 4 wherein said database-processing step comprises:

- providing the defined name and security classification category as inputs;
- obtaining a file identifier (FID) for the defined name; and
- adding the FID to DN mapping containing the security classification category to the mapping database.

Claim 6. The method as described in claim 4 wherein said searching step comprises:

- searching the FID to DN mapping database for the security classification category for the FID of the target resource; and
- returning the security classification category and defined name for the target FID, when a security classification category for the target FID was found during said search.

Claim 7. The method as described in claim 4 wherein said searching step comprises:

- searching the FID to DN mapping database for the security classification category for the FID of the target resource;

- retrieving a FID from the FID chain, when the search does not find a security classification category for the FID of the target resource;

- searching the FID to DN mapping database for the security classification category for the FID of the FID chain; and

- returning the security classification category and defined name for the target FID, when a security classification category for the target FID was found during said search.

Claim 8. The method as described in claim 7 further comprising the steps of:

- determining whether more entries are in the FID chain, when the search does not find a security classification category for the FID used in the search;

- retrieving the next FID in the FID chain; and

- searching the FID to DN mapping database for the security classification category for the currently retrieved FID of the FID chain.

Claim 9. The method as described in claim 8 further comprising the step of terminating the method when no security classification category is found for any FID in the FID chain.

Claim 10. The method as described in claim 3 wherein said flushing step comprises:

- retrieving the path name for the target resource, said path name being to a directory for the target resource;

- obtaining a vnode for the directory;

- generating a FID for the directory using the vnode;

- searching for a FID chain matching the directory FID; and

- removing FID chain from cache, when matching FID chain is found.

Claim 11. The method as described in claim 10 further comprising before said searching step the step of sorting the FID chains in the FID chain cache into hash list.

Claim 12. The method as described in claim 11 wherein said searching step comprises:

- retrieving the first FID chain in the FID chain list;
- comparing each FID in said first FID chain to said directory FID;
- determining whether there are more FID chains in the list, when said FID chain did not match said directory FID;
- retrieving the next FID chain in the FID, and
- returning to said comparing step using newly retrieved FID chain.

Claim 13. The method as described in claim 11 wherein said searching step comprises:

- retrieving the first FID chain in the FID chain list;
- comparing each FID in said first FID chain to said directory FID;
- determining whether there are more FID chains in the list, when said FID chain did not match said directory FID; and
- terminating method when no FID chain is found.

Claim 14. A computer program product in a computer readable medium for use in constructing and caching a chain of file identifiers that represent a full path to a file system resource comprising:

- instructions for processing a file system resource's defined name (DN) into a file identifier (FID) and defined name database;

- instructions for retrieving a file identifier for the file system resource that corresponds to the processed defined name of the file system resource, this file identifier being the target file identifier in the chain;

- instructions for retrieving the file identifier for the next file system resource, said next file resource being the parent of the previous file system resource in the full path;

- instructions for adding the retrieved file identifier to the chain; and

- instructions for repeating said retrieving the file identifier for the next file system resource step and said adding the retrieved file identifier to the chain step until a file identifier for each system resource in the full path of the initial file system resource in the chain.

Claim 15. The computer program product as described in claim 14 further comprising instructions for

- retrieving a file identifier corresponding to the file system resource, which is the target of an access attempt and a file identifier chain for the directory of the target system resource;

- searching for the effective security classification category and defined name for the target resource file identifier;

- updating the security classification system, when said search finds a security classification category for the target resource file identifier;

- determining whether operations for the target file system resource could affect the file system name space;

- terminating said method when operation does not affect the file system name space; and

- flushing a file identifier chain cache when there is a determination that desired operations on the target file system resource could affect the file system name space.

Claim 16. The computer program product as described in claim 15 wherein said flushing instructions comprise:

- instructions for retrieving the path name for the target resource, said path name being to a directory for the target resource;

- instructions for obtaining a vnode for the directory;

- instructions for generating a FID for the directory using the vnode;

- instructions for searching for FID chain matching directory FID; and

- instructions for removing FID chain from cache, when matching FID chain is found.

Claim 17. The computer program product as described in claim 15 wherein said searching instruction further comprises:

- instructions for searching the FID to DN mapping database for the security classification category for the FID of the target resource;

- instructions for retrieving a FID from the FID chain, when the search does not find a security classification category for the FID of the target resource;

- instructions for searching the FID to DN mapping database for the security classification category for the FID of the FID chain; and

- instructions for returning the security classification category and defined name for the target FID, when a security classification category for the target FID was found during said search.

Claim 18. The computer program product as described in claim 17 further comprising:

- instructions for determining whether more entries in the FID chain, when the search does not find a security classification category for the FID used in the search;

- instructions for retrieving the next FID in the FID chain; and

- instructions for searching the FID to DN mapping database for the security classification category for the currently retrieved FID of the FID chain.

Claim 19. The computer program product as described in claim 18 further comprising before said searching, instructions for sorting the FID chains in the FID chain cache into hash list.

Claim 20. The computer program product as described in claim 19 wherein said searching instructions comprise:

instructions for retrieving the path name for the target resource, said path name being to a directory for the target resource;

instructions for obtaining a vnode for the directory;

instructions for generating a FID for the directory using the vnode;

instructions for searching for a FID chain matching the directory FID; and

instructions for removing FID chain from cache, when matching FID chain is found.

Claim 21. The method as described in claim 2 wherein said file identifier retrieval step comprises:

retrieving the path name of the file resource, which is the target of the access attempt;

obtaining a FID for target resource with said path name;

determining whether obtained FID is in a FID chain; and

returning the target FID and FID chain, when the target resource FID was found in the FID Chain Cache.

Claim 22. The method as described in claim 21 further comprising after said path name retrieval step, the step of obtaining vnodes for the target path and parent directory.

Claim 23. The method as described in claim 2 wherein said file identifier retrieval step comprises:

- retrieving the path name of the file resource, which is the target of the access attempt;

- obtaining a FID for target resource with said path name;

- determining whether obtained FID is in a FID chain; and

- constructing a FID chain for the parent directory, when no FID chain is found.

Claim 24. The method as described in claim 23 wherein said FID chain construction comprises:

- setting a temporary vnode to equal the vnode for the parent of the target resource;

- determining whether the temporary vnode is the root directory; and

- inserting FID chain into FID chain into FID chain cache with the first FID in the chain serving as the entry search key, when temporary vnode is the root directory.

Claim 25. The method as described in claim 23 wherein said FID chain construction comprises:

- setting a temporary vnode to equal the vnode for the parent of the target resource;

- determining whether the temporary vnode is the root directory;

- retrieving a vnode for the next parent in the directory path and determining whether that parent is the root directory; and

- repeating said retrieving step until parent is the root of the directory.

Claim 26. The method as described in claim 25 further comprising the step of inserting a completed FID chain into the FID chain cache when the parent is the root directory.

Claim 27. A computer connectable to a distributed computing system, which includes file system objects containing information accessed during the execution of application and system programs comprising:

- a processor;
- a native operating system;
- application programs;
- an external authorization program overlaying said native operating system and augmenting standard security controls of said native operating system;
- a file identifier chain, which represents the full path to a target resource;
- a cache storage location for store file identifier chains which represent paths to system resources, said cache providing for faster searches of file identifiers; and
- an access decision component within said external authorization program for determining access to protected file system objects.